SOLVING PRODUCT MATCHING PROBLEM BY ENSEMBLE METHOD

[1] *Cheng-Yi Lai* (賴政毅), [2,*]*Chiou-Shann Fuh* (傅楸善)

[1] Graduate Institute of Networking and Multimedia,
National Taiwan University, Taipei, Taiwan
[2]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[*]E-mail: r09944060@ntu.edu.tw fuh@csie.ntu.edu.tw

Abstract

In this paper, we solve product matching problem by perceptual hash, EfficientNetB0 [18], TF-IDF, and BERT [19]. Then accelerate computing process by Nearest Neighbor based on *RAPIDS* library [13]. For proper parameter setting, we also analyze dataset and try different settings.

Keywords: *Hamming distance, perceptual hash, Cosine distance, EfficientNetB0, TF-IDF, BERT.*

1. INTRODUCTION

Thanks to the technology improvement, online shopping is more and more common in our life now. It is so convenient that we can enjoy shopping at home without the limitation of time and space. On the other hand, it also reduces the threshold for people who want to be a retailer or just sell used goods in house. E-commerce market has grown rapidly in recent years, especially at East-Asia. A recent study[1] estimate that the global E-commerce market size in 2019 was over USD 9 trillion, and it will be grow at a compound annual growth rate of 14.7% from 2020 to 2027. But there are still some problems in E-commerce to overcome, one of them is product matching problem. Product matching problem makes consumer spend more cost for finding target products and comparing their prices.

In this paper, we want to participate in the competition [2] held by Shopee at Kaggle and resolve product matching problem. The competition's target is to predict if two products are the same by their title, image, and perceptual hash of the image. At the same time, there are some difficulties in the competition, including: 1. Restricted to run code online so that computing resource and running time are strictly limited. 2. There's no generally common feature in data of the same product. For the limitation of computing resource and running time, we prefer to use method as simple as possible, and for the messy data, we prefer to combine different methods to predict and use ensemble method.

For these concerned, we prefer to use ensemble method to improve the performance. Generally speaking, the same items usually have similar images or title, so we can compare perceptual hash and embedding of the titles and images in order to get similarity between two items. We decide to use Hamming distance of perceptual hash and Cosine distance of EfficientNetB0 [18] vectorized image, term frequency–inverse document frequency (TF-IDF) vectorized titles and BERT vectorized titles.

2. DATA UNDERSTANDING

For proper parameter setting, we need to analyze and get the statistics of trainset.

2.1. Data visualization

We print images with the same label and check the similarity by naked eyes. It is tricky that most of the images are not similar even in the same label group, many retailers would add their trademark, advertisement words or whatever they want into image before they post, or even worse, they just take a new photograph of the product with unknown light, angle and scale (Fig. 1).

By printing out part of titles in same Label group, we can find that the diversity of title is even higher than image (Fig. 2). Most part in the title are description or specification of the product, other part are information of retailers and discount message, and the rest part are punctuation or whatever try to grab customer's notice. At the same time, these are written in different languages. There are lots of noise in title, and we can only find the similarity by the part which describe the product.

Fig. 1: Label group with the highest image diversity.

```
'ARSHOP MAYCREATE MOISTURIZING UV SPRAY 150ML (1 KG MUAT 6PCS)',
'\\xe2\\x9d\\xa4 RATU \\xe2\\x9d\\xa4 MAYCREATE MOISTURIZING SPRAY - LOT
'MAYCREATE MOISTURIZING LOTION/WHITENING SUNSCREEN SPF 50',
'Jakarta Maycreate Moisturizing Spray Melembabkan Sunblock Whitening Spr
'MAYCREATE LOTION SPRAY 150ML - Maycreat isolation protection body cream
'ISOLATION WHITENING PROTECTION SPRAY',
'MAYCREATE LOTION SPRAY 150ML - Maycreat isolation protection body cream
'ORIGINAL 100% Maycreate Whitening Spray SPF50',
'MY CREATE SPRAY PEMUTIH VIRAL TIKTOK LOTION KOREA',
'MAYCREATE MOISTURIZING SPRAY 150ML / kimberlin',
'MAYCREATE MOISTURIZING SPRAY 150ML zoomstar',
'Maycreate',
'Maycreate isolation spray',
'[IMPORT] - Maycreate Isolation Protection Spray 150ml',
'b"M\'AYCREATE MOUSTURIZING PROTECTION SPRAY /PEMUTIH KULIT"',
'MAYCREATE MOISTURIZING LOTION/WHITENING SUNSCREEN SPF 50 WATERPROOF',
'maycreate instant  ORIGINAL whitening spray / INSTANT SPRAY MYCREATE',
'Maycreate lotion spray / sunscren lotion',
'Paket dosis tinggi',
'MAYCREATE MOISTURIZING SPRAY 150ML ( 1 KG  6PCS)']
```

Fig. 2: Parts of Titles in same Label group.

2.2 Perceptual Hash Hamming Distance Distribution

We calculate the Hamming distance of perceptual hash between every product and product with same label, we discard the duplicate data and only count in unique perceptual hash to reduce the influence of data distribution on our analysis. Then we calculate Minimum Spanning Tree (MST) for products in every label group and get the maximum distance in MST (Fig. 3). It is very helpful to understand the distribution, by doing so, we can realize the minimum threshold we need to connect all product in the same label group. We can find that about half of the label group need threshold about 25 to connect all product in it (Fig. 3a, Table 1), and Hamming distance between every product takes 30.91 as the mean to present a normal distribution (Fig. 3b, Table 1). Combine two histograms of Fig. 2 and evaluate them by probability (Fig. 4), we consider 15 or below as a proper threshold. It can cover about half of the maximum Hamming distance in MST and avoid most of distance between every product, for the rest part, we use other method to connect them.

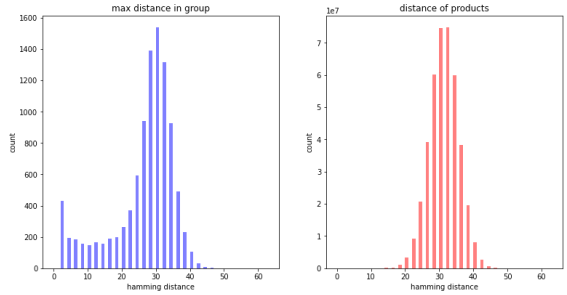

Fig. 3: (a) Histogram of maximum Hamming distance in MST of product with the same label group. (b) Histogram of Hamming distance between every product.

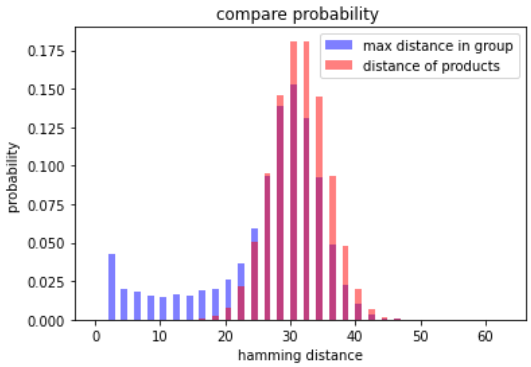

Fig. 4: Compare probability of Hamming distance between every product and maximum Hamming distance in MST of product with the same label group.

	Mean	Standard deviation
distance between every product	30.91	4.32
max hamming distance in MST of product with same label group	25.87	9.26

Table 1: Mean and Standard deviation of hamming distance between every product and maximum Hamming distance in MST of product with the same label group.

2.3. EfficientNetB0 Cosine Distance Distribution

After we extract embedding from image by EfficientNetB0, we keep top 50 similar embedding which evaluate by Cosine distance to the product. We can find that the distribution of product with the same label group is much more flatten than the distribution of top-50 similar products, but both of the distributions are centered at 0.4. And the part where distance less than 0.1 is almost overlap. (Fig. 5)

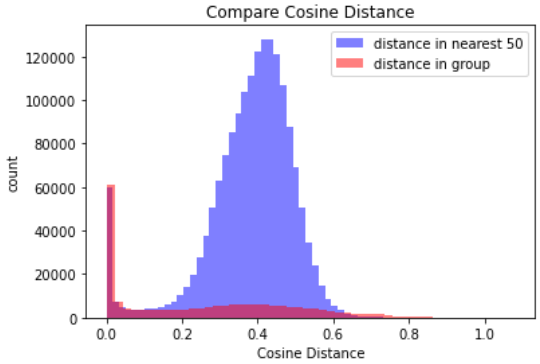

Fig. 5: Compare distribution of EfficientNetB0 vectorized image's Cosine distance between top-50 similar products and product with the same label group.

2.4. TF-IDF Cosine Distance Distribution

We keep top-50 similar embedding and then calculate both the Cosine distances of titles between every product and product with same label. Cosine distance between top-50 similar products is concentrated and centered on 0.8. And the distribution of product with same label group is so flat that it is hard to get precise prediction by thresholding. (Fig. 6)

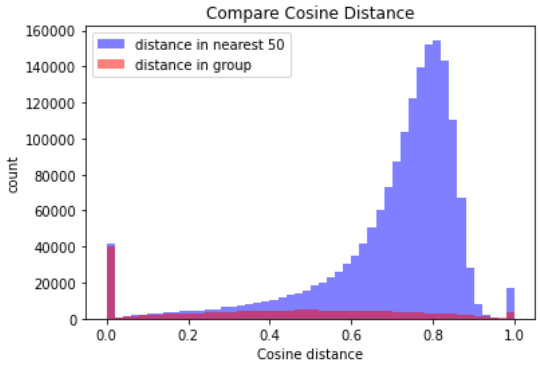

Fig. 6: Compare distribution of TF-IDF vectorized title's Cosine distance between

top-50 similar products and product with the same label group.

2.4. BERT Cosine Distance Distribution

As EfficientNetB0 and TF-IDF, we keep top 50 similar embedding and find out the distribution of their Cosine distance. The distribution of product with same label group is much more concentrate than other method. On the other hand, two distribution are centered on 0 and 0.5 relatively. Both the property make it easy to separate them. (Fig. 7)

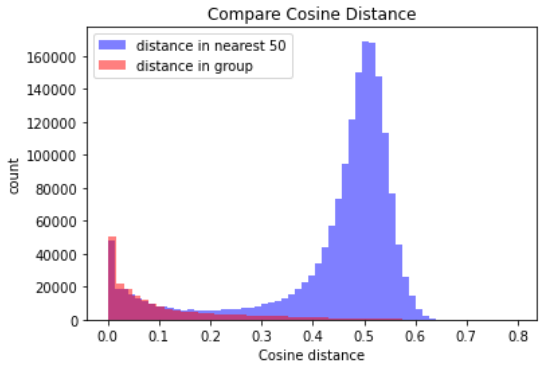

Fig. 7: Compare distribution of BERT vectorized title's Cosine distance between top-50 similar products and product with the same label group.

3. RELATED WORK

Product matching problem have been explored for a long time, and many approaches have been proposed in recent years. These approaches can be simply divided into two parts according to the topic they are focus on, including image-based method and text-based method.

3.1. Image-based method

After CNN model is widely used in image classification, A. Babenko et al. [10] propose to use the values in top layers of large CNN model as neural code for image retrieval. The Approach also improve the performance of the retrieval by retraining model with dataset which is similar to test set. At the end of the paper, author mention that the performance may be improved further with utilizing Siamese networks for image matching, which inspire the next approach.

I. Melekhov, et al. [9] train a CNN model to get neural code of image and measure the similarity by Cosine distance, when the CNN model is trained with contrastive loss function in a Siamese network. The approach first proposes to use Siamese network on whole-image matching.

Meuschke, Norman, et al[11] construct a plagiarism detection model with perceptual hashing, ratio hashing and position-aware OCR text matching, in order to compare the image similarity in academic documents. Instead of set a threshold to define the similarity degree, the approach does the job by detecting outlier of the distance.

Ristoski, Petar, et al. [12] use both image and text as feature to evaluate product similarity. The approach constructs four feature extraction models using the combination of CRF, text embedding, CNN and a dictionary-based model [4]. And they feed these features into classifiers to match products.

3.2. Text-based method

Ghani et al. [3] use Naive Bayes and a semi-supervised co-EM algorithm to extract attribute-value pair features from product description on website, and get similarity between products by comparing these attribute-value pair features. Although the approach's performance is promised by an evaluation on apparel products, there is still some constrains, the approach can only extract feature from text which is consist of both attribute name and its value (ex: texture and soft).

Kannan et al. [4] proposed a method to

match offers and products, which extract features from products description of Bing products catalog and use these features to build a semantic-based Logistic regression model offline.

D. Vandic et al. [5] present a platform to aggregate information of products from different E-shops. The platform collect RDFa annotations submitted by E-shop with ping service, then use hierarchical clustering method and several text similarity functions to match products.

Shah et al. [6] use both semantic-based classification and similarity algorithm to match product. For the concern that classification algorithm can do better than similarity algorithm for the product which is in the training dataset, but similarity algorithm can match the unseen product or label which is unsolvable for classification algorithm. Both algorithms are implemented by deep neural network, they use shallow neural network based on *fastText* [7] library for classification, and use parallel Siamese networks combined with bi-LSTM for similarity.

4. SOLUTION

We use ensemble method to cover the product with high image diversity or text diversity (Fig. 7). First, we extract image embedding from image by pretrained EfficientNetB0 and text embedding from title by TF-IDF. Then we evaluate similarity by Hamming distance of perceptual hash, Cosine distance of image embedding and Cosine distance of text embedding. To accelerate the process of getting pair-wise distance of products, we compute the distance on GPU based on K-Nearest Neighbors (KNN) in *RAPIDS* [13] library. Finally combine two predictions and generate the output file.

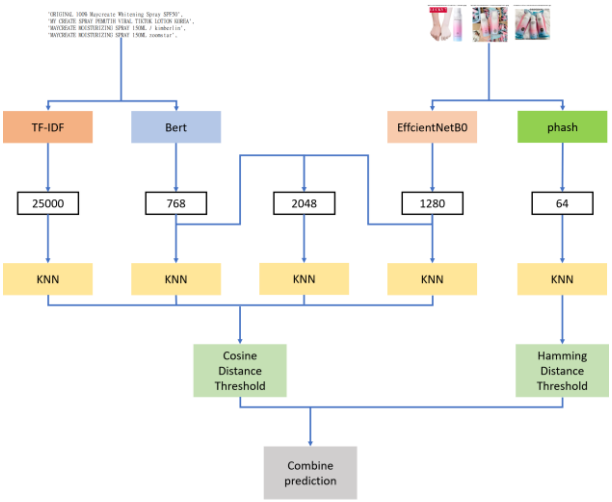

Fig. 8: Flow chart of our solution.

4.1 Perceptual Hash

Perceptual hash is an algorithm to convert multimedia to fingerprint which can preserve low-frequency signal of the image, and it is widely used in plagiarism detection and image retrieval.

Our approach includes perceptual hash for image similarity because it is more reliable and robust than other hash function like average hash or difference hash, perceptual hash also shows less collisions. The robustness can be showed especially when doing gamma correction or histogram matching on image, we still can get similar hash. It is a suitable and cheap way converting image to fingerprint or embedding [14-17].

4.2. EfficientNetB0

EfficientNet is a series of structures proposed by Google at 2019 [18]. The new structures and experiment proved efficient hyperparameter make it fast and accurate. EfficientNetB0 is the structure proposed as the baseline of the approach, and the structure is searched by MnasNet. We use the pretrained EfficientNetB0's final layer as image embedding because its parameter number is the smallest one in EfficientNet family, and the execute

speed is the fastest one, too. It enables us to concentrate on turning hyperparameters like threshold of similarity.

4.3. TF-IDF

Term Frequency–Inverse Document Frequency (TF-IDF) is a useful method and commonly used in information retrieval. It also been used to evaluate image similarity in Computer Vision like Simultaneous localization and mapping (SLAM) [20]. The method is based on the assumption that the most informative words in the document must be those words appear frequently in the document and appear less frequently in other document. Although the drawback of TF-IDF have been indicated that the assumption is not totally correct and it cannot extract position information and semantics, we hope to we can extract the important information of title, just like product specification or description.

4.4. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a famous and popular model in Natural Language Processing region for its impressive performance and convenience. BERT can not only extract semantic in sentence but also semantic between sentence, we hope the property can extract semantic from the special structure of titles, actually, BERT have been used in many approaches for product matching [21, 22]. We use BERT pretrained on Indonesian language because it worked the best on the competition dataset, and fine-tune the model with 80% of competition public dataset.

4.5. Threshold and Combine Prediction

We evaluate perceptual hash with Hamming distance and evaluate embeddings with Cosine distance, and get top-50 similar product by Nearest Neighbor based on RAPIDS library [13], because the competition has been guaranteed that group size is at most 50. Other than NN model, we also use several methods in order to get more precise prediction, including setting proper threshold, concatenating embeddings, and combining these predictions by simple union.

We reference the distribution in Sec. 2 and keep trying different setting to get the proper threshold. During the tuning process, we find that getting precision prediction is much more important than getting more prediction from single method. Because we will union prediction from several method at the end, not included matching in single method may be covered by the prediction of other method. But it is tricky to get rid of the false positive prediction, thus we tend to set a strict threshold.

Concatenating the EfficientNet embedding and BERT embedding is our another attempt to cover the missing of strict threshold policy. Considering both image and text similarities simultaneously may find out the matching that either image similarity or text similarity is closely but not high enough to pass the threshold. Actually, discussion in the competition [23] declare the method can efficiently improve the performance.

5. RESULTS

We recorded the running time and their performance of methods in Table 2. The experiment was performed on the NVIDIA TESLA P100 GPU provided by Kaggle. We also use the methods in Sec. 4 to predict respectively, and try all possible combinations of them, the f1-score of training dataset and testing dataset are shown in Table 3.

Through Table 2, we can find that the performance of the graphical method is

slightly lower than that of the text method when using the training dataset, but the performance is very similar when using the test dataset. When we focus on running time, the graphical method is much better than text method, the fastest one and the slowest one has a huge gap of more than 10 times in running time. Compared with more accurate DNN methods such as BERT and EfficientNet, we believe that the advantages in speed and computational efficiency are the reason why more "simple" methods such as hash and TFIDF have not been eliminated so far.

The f1-score of each method in the training dataset shows close relation to the distribution in Sec. 2. More separable the distributions are, higher f1-score we get in the training dataset, just like what BERT and EfficientNetB0 have done. The result is predictable, because the dense distribution in group enables to set bigger threshold and include more matching. Another result is unpredictable that the f1-score in training dataset and testing dataset are not positively correlated. It may be due to overfitting caused by threshold tuning, or there is critical different distribution between training dataset and testing dataset. As a result, the best combination in testing dataset is TF-IDF, EfficientNetB0, the embedding mixed by EfficientNetB0 and BERT.

	TFIDF	phash	EFFNet	BERT
Time (second)	50.75	33.89	340.42	473.16
F1-score (trainset)	0.591	0.587	0.659	0.829
F1-score (testset)	0.589	0.59	0.65	0.622

Table 2: Running time of each method on the NVIDIA TESLA P100 GPU evaluate by second.

TFIDF	phash	EFFNet	BERT	Mix	trainset	testset
V	V	V	V	V	0.849	0.676
V	V	V	V		0.849	0.676
V	V	V		V	0.743	0.707
V	V	V			0.727	0.707
V	V		V	V	0.845	0.676
V	V		V		0.845	0.66
V	V			V	0.736	0.707
V	V				0.680	0.672
V		V	V	V	0.680	0.677
V		V	V		0.850	0.677
V		V		V	0.744	0.708
V		V			0.728	0.708
V			V	V	0.830	0.677
V			V		0.824	0.622
V				V	0.720	0.708
V					0.591	0.589
	V	V	V	V	0.853	0.677
	V	V	V		0.853	0.677
	V	V		V	0.692	0.649
	V	V			0.658	0.649
	V		V	V	0.850	0.677
	V		V		0.849	0.661
	V			V	0.686	0.649
	V				0.587	0.59
		V	V	V	0.854	0.678
		V	V		0.854	0.678
		V		V	0.693	0.65
		V			0.659	0.65
			V	V	0.835	0.678
			V		0.829	0.622
				V	0.669	0.65

Table 3: All combinations of methods where testset score is the public score of competition.

6. REFERENCES

[1] Grand View Research (2020), E-commerce Market Size, Share & Trends Analysis Report By Model Type (B2B, B2C), By Region (North America, Europe, Asia Pacific, Latin America, Middle East & Africa), And Segment Forecasts, 2020 – 2027

[2] Kaggle competition, Shopee - Price Match Guarantee, from https://www.kaggle.com/c/shopee-

product-matching/overview (May 19, 2021)
[3] Ghani, Rayid, et al. "Text mining for product attribute extraction." *ACM SIGKDD Explorations Newsletter* 8.1 (2006): 41-48.
[4] Kannan, Anitha, et al. "Matching unstructured product offers to structured product specifications." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011.
[5] Vandic, Damir, Jan-Willem Van Dam, and Flavius Frasincar. "Faceted product search powered by the Semantic Web." *Decision Support Systems* 53.3 (2012): 425-437.
[6] Shah, Kashif, Selcuk Kopru, and Jean David Ruvini. "Neural network based extreme classification and similarity models for product matching." *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. 2018.
[7] Joulin, Armand, et al. "Bag of tricks for efficient text classification." *arXiv preprint arXiv:1607.01759* (2016).
[8] Babenko, Artem, et al. "Neural codes for image retrieval." *European conference on computer vision*. Springer, Cham, 2014.
[9] Melekhov, Iaroslav, Juho Kannala, and Esa Rahtu. "Siamese network features for image matching." *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016.
[10] Babenko, Artem, et al. "Neural codes for image retrieval." *European conference on computer vision*. Springer, Cham, 2014.
[11] Meuschke, Norman, et al. "An adaptive image-based plagiarism detection approach." *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. 2018.
[12] Ristoski, Petar, et al. "A machine learning approach for product matching and categorization." *Semantic web* 9.5 (2018): 707-728.
[13]RAPIDS library, from https://rapids.ai/ (May 26, 2021)
[14] Hadmi, Azhar, et al. "Perceptual image hashing." *Watermarking-Volume 2*. IntechOpen, 2012.
[15] Jiaheng, Huang, et al. "A comparative study on image similarity algorithms based on hash." *Journal of Dali University* 2.12 (2017): 32.
[16] Srivastava, Siddharth, Prerana Mukherjee, and Brejesh Lall. "imPlag: Detecting image plagiarism using hierarchical near duplicate retrieval." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.
[17] Testing different image hash functions, from https://content-blockchain.org/research/testing-different-image-hash-functions/ (May 27, 2021)
[18] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International Conference on Machine Learning*. PMLR, 2019.
[19] Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
[20] Thrun, Sebastian. "Simultaneous localization and mapping." *Robotics and cognitive approaches to spatial mapping*. Springer, Berlin, Heidelberg, 2007. 13-41.
[21] Tracz, Janusz, et al. "BERT-based similarity learning for product matching." *Proceedings of Workshop on Natural Language Processing in E-Commerce*. 2020.
[22] Peeters, Ralph, Christian Bizer, and Goran Glavaš. "Intermediate training of BERT for product matching." *small* 745.722 (2020): 2-112.
[23]Discussion in Shopee competition, from https://www.kaggle.com/c/shopee-product-matching/discussion/238136 (June 11, 2021)